**Appendix One - Coursework Coversheet**

|  |  |  |  |
| --- | --- | --- | --- |
| **School of Geography**    **FACULTY OF ENVIRONMENT** | | logo_black.gif | |
| **Student ID** | 201375952 | **Word count** |  |
| **Module title / code** | GEOG5990M | **Mark**  **Less deduction**  **(state reason)**  **Final Mark** |  |
| **Assignment title** | Assignment 2 – Project Report |
| **Marker** |  |

|  |
| --- |
| **To improve your work for next time:** |
| 1. |
| 2. |
| 3. |

|  |
| --- |
| **Justification of mark (using specific text from criteria)** |
|  |

|  |
| --- |
| **Additional comments** |
|  |

**Programming to Aid in Decision Making: Location of a factory**

Contents

[**Appendix One - Coursework Coversheet** 1](#_Toc72411019)

[**Introduction** 2](#_Toc72411020)

[**Study Area/ Geographical Aspect** 2](#_Toc72411021)

[**Data and Processing** 2](#_Toc72411022)

[**Results and discussion** 3](#_Toc72411023)

[Geology 3](#_Toc72411024)

[Transport 3](#_Toc72411025)

[Population 4](#_Toc72411026)

[Combining/Merging the datasets 5](#_Toc72411027)

[Hypothetical Scrollbar values 5](#_Toc72411028)

[Model Implementation 5](#_Toc72411029)

[**Conclusion** 6](#_Toc72411030)

[**References** 7](#_Toc72411031)

# **Introduction**

Computer programming is a series of instructions often in English but rather simpler to fathom, written to a computer (Langtangen, 2014) telling it what to do. This in essence simplifiers how tasks, workflows can be undertaken. This project leverage python a high-end programming language to undertake a geographical analysis scenario of identifying a suitable location for a rock aggregate factory in the UK. This entails developing a program that reads in the data makes some display writes output files.

# **Study Area/ Geographical Aspect**

This is a national level project for the UK, evaluating parameter across the whole country except Northern Ireland.

# **Data and Processing**

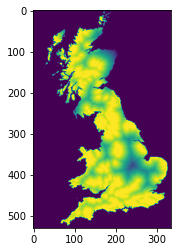
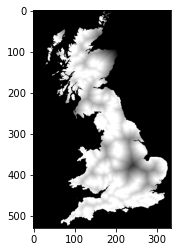
Three raster datasets representing geology, transport network and population were used in the exercise. The data was converted to readable .txt format and subsequently loaded in the program. This was then opened using the keyword *with* that sets up a context manager instructs how the code runs and closes the file once the clause is left. This remedy the closing of file manually.

# **Results and discussion**

All the files were read in one at a time and displayed.

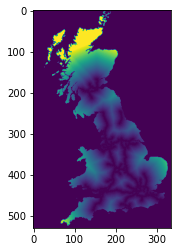
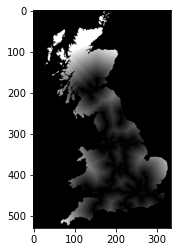
## Geology

The function was run, and a display map shown below was obtained with the lighter values showing the areas with good geology whereas the dark areas are the ones considered to have unsuitable geology.

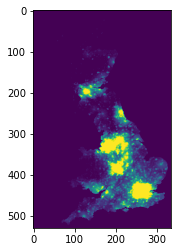
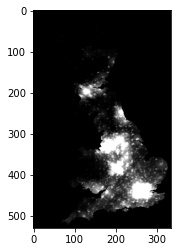
## Transport

Similarly, the program generated the values of the transport network as show below. The darker colours showing the areas with the best transport network. The areas towards the north of UK in Scotland show some areas with poor road network that could be attributed to the forest in the area. The rest of the areas show a very well-established transport network and uniformly distributed.

## Population

Lastly, the population data was read using the same function as shown in the maps below. This shows a very telling pattern with city centres like London and other urban centres exhibiting the highest population patterns.

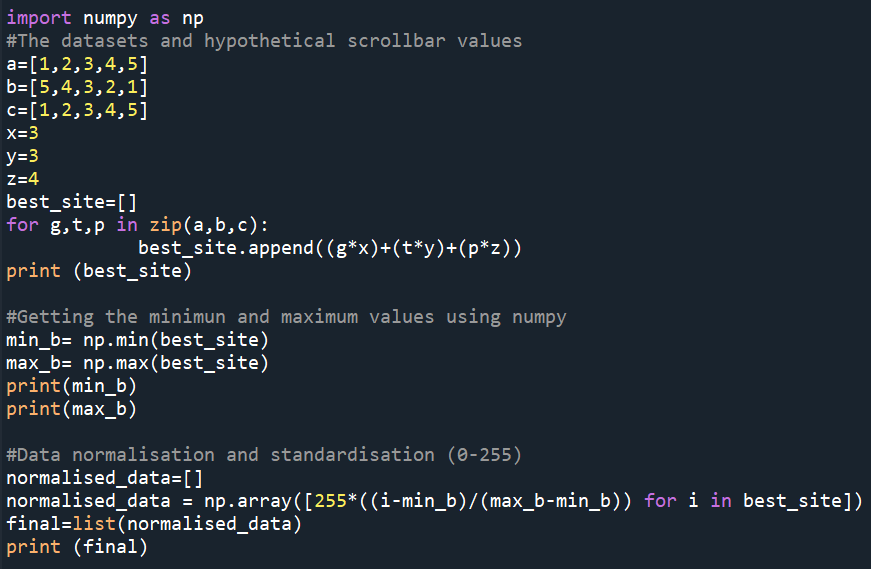
 

## Combining/Merging the datasets

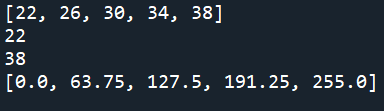
To achieve this, pseudo lists were prepared to be scaled up in the main model as demonstrated below. This is the equivalent of writing the problem done programmatically.

### Hypothetical Scrollbar values

Hypothetical values for weighting the different layers are adopted to and variables x, y and z assigned the values. The merge is implemented using a for loop that picks elements from each list applies a scrollbar value and generates a combined dataset

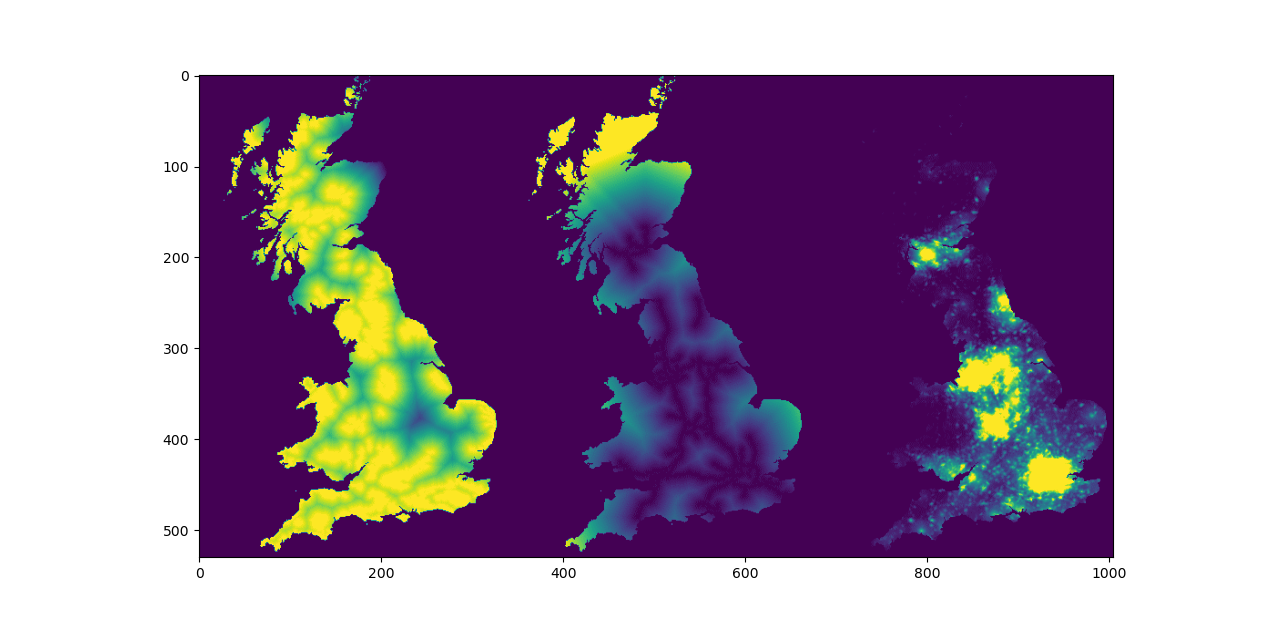


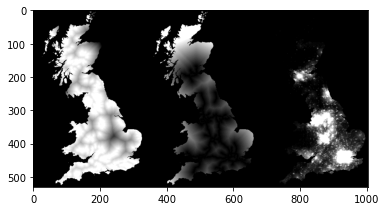
Results from the pseudo model



### Model Implementation

The model was implemented as per the above, however the merging produced three images as shown below instead of the desired combined image. This was due to the model inefficacy and an attempted to resolve was attempted in the agentframework module in the sophisticated model however this was equally resulting in the same output.





# **Conclusion**

The project has demonstrated the ability of using python to implement a location intrinsic problem by weighting the various input parameters. This model suggests a very efficient approach to automating the process. The obtained results are inconclusive and adjustments to the model are necessary for it to function optimally.

# **References**

Hans Petter Langtangen. 2014. *A Primer on Scientific Programming with Python (4th. ed.).* Springer Publishing Company, Incorporated.

McKinney Wes 2012. *Python for data analysis (1st. Ed).* O’Reilly Media, Inc

Ramalho, L. (2015), *Fluent Python:* *Clear, Concise, and Effective Programming* , O'Reilly Media .